Business Academy SouthWest
Lecturer Kristian Wittrup & Lecturer Søren Spangsberg Jørgensen
2022

PWA & REST API project description

groupAssignment_allowed : “yes”
maxMemberOfGroup : 3
handInData(tentative) : 24th of May, 2022

Introduction

Main task: Design & develop a project management application.

You could get inspired by Trello, KanBan, Google Sprint, SCRUM, etc.
Or you could set the main purpose of the application usage to be built around when you/group would start up a new project. Fx course- or semester projects. You could give a choice at start of creation, for a “clean” project, or set up a “default” project, that comes with a set of “standard tasks” already (ie. *report, research phase, project management, SEO, etc.*).

- What could a “standard items/task” be?
  - How would you set this up? (hard-coded into Vue, data from REST API??)

- Or would you create a clean project with no “standard items” (*ie pre-programmed template*)

You need to think about the structure, and how you want to store and handle your data and users.

- Data: Own REST-API, Mongo dB, Express, Heroku, etc.
  - How do you want to store data? in in arrays, objects, references etc.? - It must be online retrieved data (*cloud based*)
  - Projects, tasks (in projects?), assigned team-members etc. – how do you want to handle this?
  - Tracking projects: hours assigned, hours used, hours left, hours allocated to team-member, warning/alert when deadline is close or remaining hours are low etc.
- Users: How to handle login, assigning users to projects, etc.
  - Do you want to add stakeholders (like in SCRUM), etc. to it as well?

- A 70-80% finished sketch/design is available on Moodle, in the assignment folder for inspiration or you can copy it 100% for the design and flow.

- UX: How is the user supposed to navigate through everything. Keep UX in mind when designing and building you project.

- The structure of the code and the design patterns of these, will be taken into consideration when evaluating. Several components (or composition functions) are better than a few very large components.

Project Requirements - must include

Project overview info

- Create either a MoSCoW model, Requirement Specification or User Stories
- Be very specific with the needs, ie: functionality for each component etc.
- This is the only document you need to hand-in on WiseFlow.
- Added bonus could be: flow chart / DB diagram(UML, …) / eXperience Design (sketch) etc.

Front-end (VueJS or likewise)

- **Component Based**
  - Divide code into smaller components: navigation, login, create project, edit, etc
  - Try to create nested components or Composables (composition functions if using Vue3 Composition API) or if using VueX, use it as your logical center (you can divide VueX up as well, into files with modules, mutation, actions etc).

- **Use Navigation guards** to show different content, depending on if the user is logged in (as admin, leader, etc) or not

- **Login system** for the administration. You can make your own, use Auth2 (Firebase) or the JWT login tokens(SMSJ video).

- **Data and logic** should use either Composition API or Vuex state management to control the data and consistency.

Back-end (REST API)

- Designed and developed in a web application framework of **your** choice.
- Full CRUD routes (Create, Retrieve, Update and Delete). These must work on data from a database (not local data).
- You are free to choose your underlying database paradigm (RDBMS, NoSQL etc.)
- User authentication on selected or all routes (registration and login)
- Test your endpoints using an appropriate program (Postman, Insomnia etc.)
- Integrated with Swagger including endpoint documentation.

Live version + GitHub (hand-in)

- You must build your project and upload it to a live site and link it in the document
- GitLab Pages, Netlify, Firebase, Heroku, old fashioned domain or somewhere else
  - Link on how to: https://cli.vuejs.org/guide/deployment.html#general-guidelines
- You must upload and link to the VCS repository

NOTE:
If you plan on using other frameworks like Angular, React, Svelte, Laravel, Ruby on Rails, .Net, Django etc. – get it approved by the teacher before you start. The same requirements apply, but with the chosen framework (with alternative the fit with it).